FINAL REPORT

Principle Investigator: R. I. Iyer
PI Institution: University of Illinois at Urbana-Champaign
PI Phone Number: (217) 333-9732
PI E-mail Address: iyer@crhc.uiuc.edu
Contract Title: Measurement-Based Dependability Evaluation of Multiprocessor Systems
Contract Number: N0014-91-J-1116
Reporting Period: Final Report

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

I. Summary of Technical Progress

In the previous reporting period, we showed that, when many users run the same software, such as an operating system, the majority of field software failures are rediscoveries of known problems. This clearly shows that the number of faults identified in a software system is not the only important factor; rediscoveries can seriously degrade software dependability. The effects of rediscoveries are: 1) more failures than the number of faults, 2) wasted service resources due to repeated data collection and diagnosis of the same problem, and 3) being unable to provide service to users until diagnosis is complete even if a solution already exists. To address the second and third effects, we developed symptom-based diagnosis of rediscovered software failures and showed that the majority of rediscoveries can be automatically diagnosed based on symptoms, such as procedure call trace and problem detection location.

A. Diagnosis of Rediscovered Software Failures

During the current reporting period, we extended the previous work in several directions. First, we evaluated the effectiveness of data-oriented symptoms, such as register values, selected local and global variables, and parameters passed between procedures, for the diagnosis of rediscoveries. For this evaluation, we restored processor memory dumps for past field software failures in Tandem systems from the archive. The results show that data- oriented symptoms are generally not as effective in rediscovery diagnosis as code-oriented symptoms, such as procedure call trace and detection location. Among data-oriented symptoms studied, register values were the most useful. Second, we evaluated the effectiveness of symptom-based diagnosis using more failure data. The results consistently showed that we can diagnose the majority of rediscoveries based on these symptoms. Also, using the restored processor memory dumps, we demonstrated that machine code detection location can be used as a signature for problem detection location. Third, we investigated the symptoms of nonsoftware failures and showed that we can separate software and nonsoftware failures based mainly on problem detection location. Finally, based on all these results, we made a tradeoff between successful diagnosis and misdiagnosis and finalized the symptom-based diagnosis strategy.

The diagnosis strategy was or is being implemented in two locations at Tandem: 1) in individual user systems (as an operating system function) and 2) at the service center (as a central diagnosis tool). The Tandem Failure Data System (TFDS)

19970806 086

Measurement Based Dependability evaluation of multiprocessor Systems

PLEASE CHECK THE APPROPRIATE BLOCK BELOW:

-AQ# M97-11-5898

□ ________ copies are being forwarded. Indicate whether Statement A, B, C, D, E, F, or X applies.

☒ DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED

□ DISTRIBUTION STATEMENT B:
DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES ONLY: (Indicate Reason and Date). OTHER REQUESTS FOR THIS DOCUMENT SHALL BE REFERRED TO (Indicate Controlling DoD Office).

□ DISTRIBUTION STATEMENT C:
DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES AND THEIR CONTRACTORS; (Indicate Reason and Date). OTHER REQUESTS FOR THIS DOCUMENT SHALL BE REFERRED TO (Indicate Controlling DoD Office).

□ DISTRIBUTION STATEMENT D:
DISTRIBUTION AUTHORIZED TO DoD AND U.S. DoD CONTRACTORS ONLY; (Indicate Reason and Date). OTHER REQUESTS SHALL BE REFERRED TO (Indicate Controlling DoD Office).

□ DISTRIBUTION STATEMENT E:
DISTRIBUTION AUTHORIZED TO DoD COMPONENTS ONLY; (Indicate Reason and Date). OTHER REQUESTS SHALL BE REFERRED TO (Indicate Controlling DoD Office).

□ DISTRIBUTION STATEMENT F:
FURTHER DISSEMINATION ONLY AS DIRECTED BY (Indicate Controlling DoD Office and Date) or HIGHER DoD AUTHORITY.

□ DISTRIBUTION STATEMENT X:
DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES AND PRIVATE INDIVIDUALS OR ENTERPRISES ELIGIBLE TO OBTAIN EXPORT-CONTROLLED TECHNICAL DATA IN ACCORDANCE WITH DoD DIRECTIVE 5230.25. WITHHOLDING OF UNCLASSIFIED TECHNICAL DATA FROM PUBLIC DISCLOSURE. 6 Nov 1984 (Indicate date of determination). CONTROLLING DoD OFFICE IS (Indicate Controlling DoD Office).

□ This document was previously forwarded to DTIC on ____________ (date) and the AD number is ________________.

□ In accordance with provisions of DoD instructions, the document requested is not supplied because:

□ It will be published at a later date. (Enter approximate date, if known).

□ Other. (Give Reason)

DoD Directive 5230.24, "Distribution Statements on Technical Documents," 18 Mar 87, contains seven distribution statements, as described briefly above. Technical Documents must be assigned distribution statements.

R.K. Iyer
Print or Type Name

217-333-7774
Telephone Number

R.K. Iyer
Authorized Signature/Date

TOTAL P.03

DTIC QUALITY INSPECTED 4

(Joyce Chiras)

is an operating system product that implements the diagnosis function. Given a failure, the TFDS extracts failure symptoms, such as procedure call trace and problem detection location. It also determines whether the failure has occurred in that system before, based on the database of all previous failures on the system. If the TFDS concludes that the failure is a rediscovery, it restarts the failed software component without collecting failure data; otherwise, it restarts the failed component after collecting failure data. The TFDS then sends the information about the incident and the results of the diagnosis to the service center. The diagnosis tool at the service center handles rediscoveries across all systems. In contrast to the TFDS, it runs in semi-automatic mode. Given a failure alarm from TFDS, it extracts all failures that have symptoms similar to the newly reported one. The tool also provides a measure of similarity between the newly reported failure and each extracted failure. Based on such information and after more investigation, human analysts can conclude whether the failure is a rediscovery.

This work is significant in that it makes it possible to avoid wasting service resources in handling rediscoveries manually and it improves service response time for rediscoveries. An important side benefit of the work is that it provides an on-line infrastructure to measure error detection and recovery capabilities of individual products. What we can measure, we can often improve. The work also brings out the need for understanding the reasons for re-occurrences and their significance, which requires additional measurement. Such measurements can guide us through the next cycle of the design and enhancement of tools as well as of software development and service methods.

The diagnosis work has deepened our understanding of software faults and their symptoms and allowed us to build software fault and error models. Such models have been used for designing fault-injection experiments as well as tools (FTAPE, described below, and DEFINE) to study the propagation and effect of faults on actual systems in a more focused and systematic manner.

B. Summary of FTAPE

FTAPE (Fault Tolerance and Performance Evaluator) is a fault-injection tool used for fault tolerance benchmarking and the study of error propagation in fault-tolerant systems. The tool utilizes stress-based injection, which is a technique to maximize the level of fault-tolerant activity in the system being tested. Stress-based injection relies on the monitoring of system resource activity to provide feedback to the fault injector to determine the fault model, time, and location for future injections.

The FTAPE fault injector is based on a fault-injection pseudo device driver that is added to the operating system. The use of a pseudo device driver allows the fault injector great flexibility in inserting faults into any software-addressable location, including main memory, CPU registers, and the disk system. In addition, because the pseudo device driver executes at the operating system level and not at the user level, conflicts with memory and resource protection are avoided, which permits injection of faults into the operating system itself.

FTAPE has been implemented on several fault-tolerant systems, including the Tandem Integrity S2 and the Tandem Integrity S4000, which is based on ServerNet technology. The tool has been used to compare the dependability of these systems, as well as the performance degradation suffered under fault conditions. Experiments with the tool have also been performed to determine the effects of concurrent processes, such as paging and memory sharing, on error propagation and detection.

II. Transitions

The symptom-based diagnosis is being implemented at Tandem Computers Inc. Dr. Inhwan Lee, who participated in the development of the method while he was at the University of Illinois, is leading this project at Tandem. The project was selected as one of the top five development projects by the Tandem system users last year. The Tandem Failure Data System (TFDS), which is an operating system product and release vehicle of the diagnosis function, was released to the field last year. The diagnosis tool for the service center is currently being built. Inhwan Lee also extended the symptom-based method so that it can be used for providing preventive software service, thus reducing the number of rediscoveries in a cost-efficient manner.

In a joint effort with Lucent Technologies, FTAPE was implemented on the STRATUS Continuum platform. First experiments were conducted to evaluate the fault-tolerant architecture. FTAPE was then redesigned to fit the needs of distributed computing (NFTAPE - Network FTAPE). This ongoing process included the development of a communication model, which allows different parts of FTAPE to interact with each other. The implemented approach is message-based and uses TCP/IP sockets. A prototype of NFTAPE was implemented in a distributed fault-tolerant environment consisting of both Sun and Windows-NT workstations at Lucent Technologies.

III. Publications
----

[1] R. K. Iyer and I. Lee,
"Measurement-Based Analysis of Software Reliability,"
Chapter 8, McGraw-Hill Handbook of Software Reliability Engineering,
McGraw-Hill, 1996.

[2] I. Lee and R. K. Iyer,
"Software Dependability in the Tandem GUARDIAN System,"
IEEE Trans. Software Engineering, Vol. 21, No. 5, pp. 455-467,
May 1995.

[3] I. Lee, G. Pitt, and R. K. Iyer,
"Efficient Service of Rediscovered Software Problems,"
Proc. Int. Symp. Fault-Tolerant Computing,
Sendai, Japan, pp. 348-352, June 1996.